IEEE Aerospace Conference 2018

Assurance of Model Based Fault Diagnosis

Allen Nikora, Priyanka Srivastava, Ksenia Kolcio, Lorraine Fesq, Seung Chung

Presented By: Priyanka Srivastava

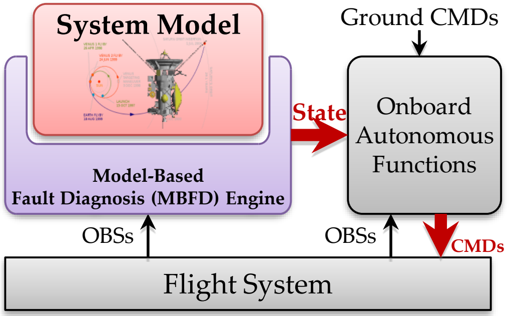
Flight-Systems Systems Engineering

Jet Propulsion Laboratory
California Institute of Technology

© 2018. California Institute of Technology. Government sponsorship acknowledged.

Assurance of Model-Based Fault Diagnosis

The Need for Reliable Onboard Model-Based Fault Diagnosis

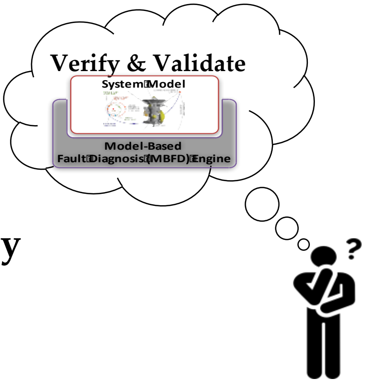

System Model
Ground CMDs
State
Onboard Autonomous Functions
Model-Based Fault Diagnosis (MBFD) Engine
OBSs
OBSs
CMDs
Flight System

What is MBFD?

- **Model-based fault diagnosis** (MBFD) can enable on-board autonomy by continuously verifying correct hardware behavior in addition to diagnosing symptoms to estimate the health state. An onboard autonomy capability can then use the determined health states to decide how to react.

The Problem:

- **No proper V&V of MBFD…**
  - Techniques for adequately verifying and validating MBFD technologies are not well understood



Our Solution:

- **Establish a concrete methodology to Verify and Validate MBFD technology**
- **Apply the V&V Methodology to an Onboard System**
  - **Model the Onboard System using MBFD technique**
  - **Conduct V&V tests on the modelled system**
- **Analyze the V&V test results to ensure the proper functioning of MBFD on the modelled system**

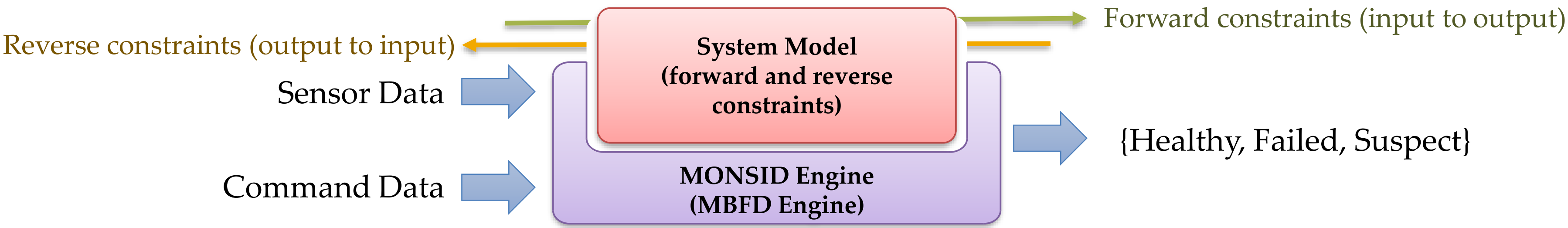
Assurance of Model-Based Fault Diagnosis

IEEE Aerospace Conference 2018

- Overview of Model Based Fault Diagnosis (MBFD)
- Approach towards Assurance of MBFD techniques
- Results
- Future Work

Model-Based Fault Diagnosis Architecture

Overview of the MONSID System and Diagnosis Engine


Forward constraints (input to output)
Reverse constraints (output to input)
Sensor Data
Command Data
System Model (forward and reverse constraints)
MONSID Engine (MBFD Engine)
{Healthy, Failed, Suspect}

The Model-based Off-Nominal State Isolation and Detection (MONSID) system is an implementation of constraint suspension extended to electro-mechanical systems, and is currently being developed by Okean Solutions. It has been prototyped in the MATLAB/Simulink environment and a C++ version is intended for deployment.

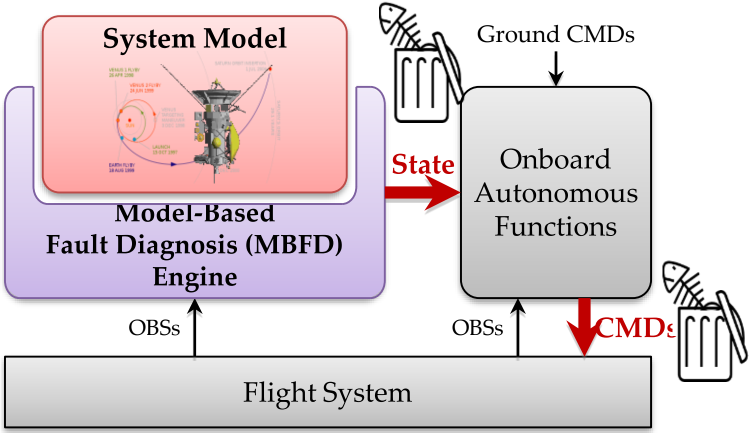
Two Main Parts

- **System Model (Diagnostic Model)**
  - Model capturing nominal system behavior
  - User defined, application specific
  - Fault models not needed
  - Data is propagated through the Model via forward(input to output) and reverse (output to input) constraints

- **Diagnosis Engine**
  - Not application specific
  - Diagnoses faults from user-supplied models of the system given measurement and command data
  - Compares forward and reverse constraint values with component boundaries (nodes) to detect any faults/inconsistencies using constraint suspension technique

Assurance of MBFD Techniques

Need for MBFD V&V Techniques



System Model

Model-Based Fault Diagnosis (MBFD) Engine

State

Ground CMDs

Onboard Autonomous Functions

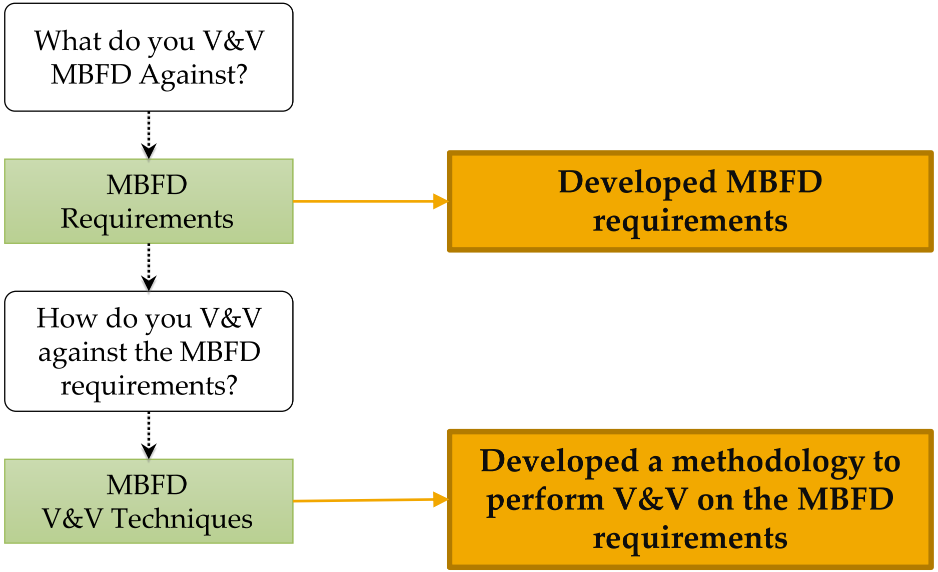
OBSs

OBSs

CMDs

Flight System

Reliable diagnosis of the health state of the system is the key to reliable onboard autonomy. Model-based fault diagnosis (MBFD) is fundamental and foundational to enabling system-wide autonomy.



What do you V&V MBFD Against?

MBFD Requirements

How do you V&V against the MBFD requirements?

MBFD V&V Techniques

Developed MBFD requirements

Developed a methodology to perform V&V on the MBFD requirements

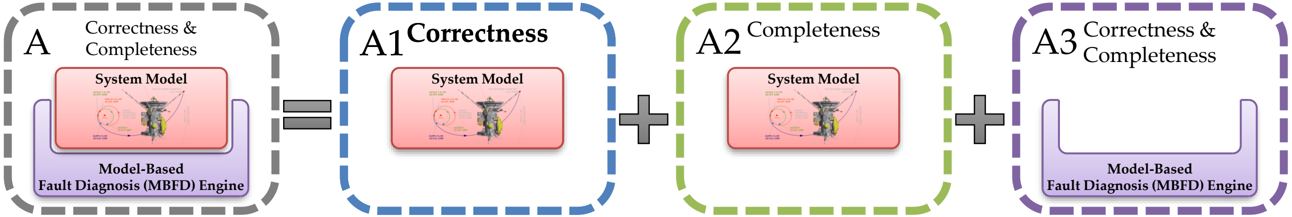
Assurance of MBFD Techniques

The Approach : V&V Methodology

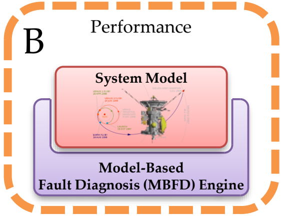
Develop and demonstrate V&V techniques that ensures correct diagnosis of system health state by MBFD.

A. Develop and demonstrate the techniques for checking the correctness and coverage/completeness of MBFD.

1. **Model Correctness** – Is the model sufficient for correct faults diagnosis?
2. **Model Completeness** – Is the model sufficient for the required fault coverage?
3. **Diagnosis Engine Correctness and Completeness** – Will the diagnosis engine diagnose all identified faults correctly and completely, given a correct and complete system model?


A
Correctness & Completeness
System Model
Model-Based Fault Diagnosis (MBFD) Engine
=
A1
Correctness
System Model
+
A2
Completeness
System Model
+
A3
Correctness & Completeness
Model-Based Fault Diagnosis (MBFD) Engine

B. Develop and demonstrate the techniques for analyzing the **performance** characteristics of MBFD. – Memory footprint and processing time, diagnostic resolution, rate of false-positives and false negatives

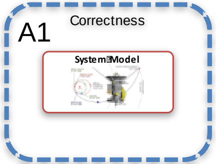


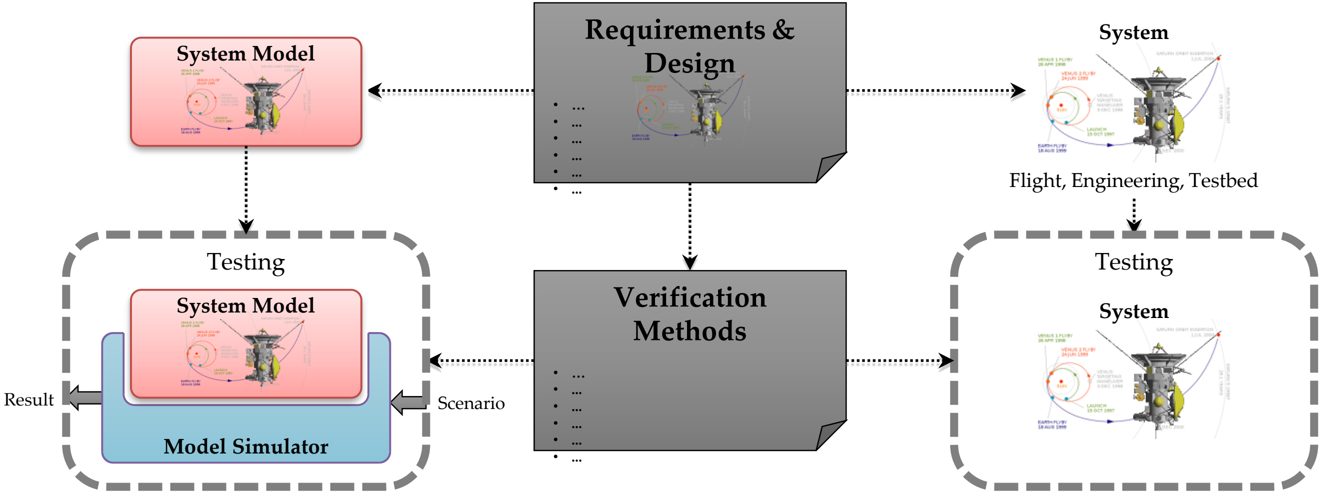
Model Correctness

V&V Testing Methodology for Model Correctness

A1. Model Correctness – Testing

- Build diagnostic system model based on functional requirements,
- Verify the model by verifying the functional requirements through tests
- Validate and update the model by analyzing and comparing the operational data




System Model

Requirements & Design

System

Flight, Engineering, Testbed

Testing

System Model

Model Simulator

Result

Scenario

Verification Methods

Testing

System

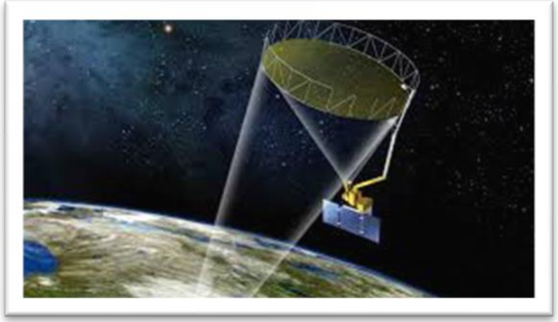
Unlike the design and verification methods of traditional monitors, the correctness criteria are well defined and made explicit by the system functional requirements.

Model Correctness

Developed testable diagnostic models

SMAP GNC System: RWA, SRU, MIMU

- **GNC functionality and behavior similar from mission to mission**
  - IMU, SRU, RWA, and other components are physically similar and perform similar functions.
  - Overall GNC system-level function remains the same – point and stabilize spacecraft during flight.
- RWA, MIMU, and SRU together form a simple GNC core system that can be extended during future work by including additional components (e.g., RCS, MTR, CSS).
- Fault protection is an important GNC element – need to respond to anomalous conditions to preserve/recover spacecraft capability.

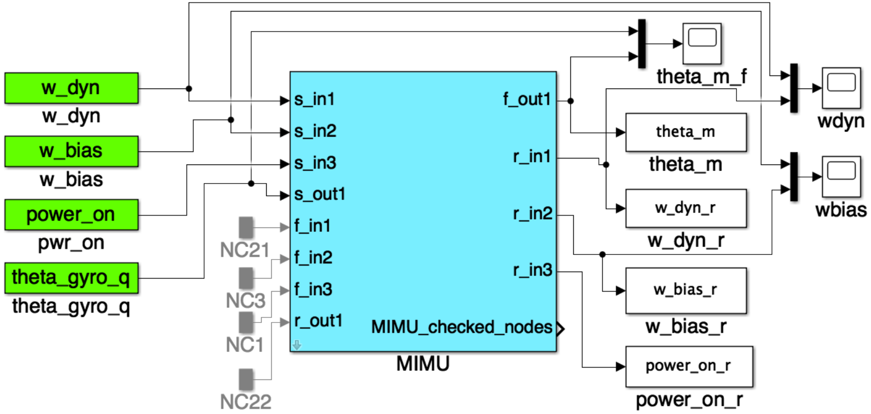


Demonstration Target – SMAP

- SMAP (Soil Moisture Active Passive)
  - Earth science mission - measure and map Earth's soil moisture
    and freeze/thaw state to better understand terrestrial water,
    carbon and energy cycles.
  - Launched 2/2015.
- Rationale
  - Developed at JPL – Provided access to all the artifacts required for this task
  - Recently developed and flown
    - Representative of current system architectures, development practices.

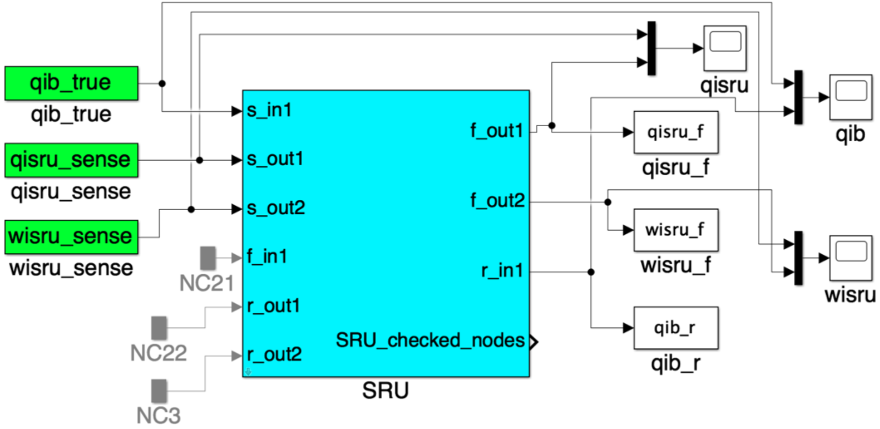
MONSID GNC Models

Component Level Models : Sensors



MONSID MIMU Model

The MONSID MIMU Model outputs 3-axis S/C angular position at 200Hz. (Note that only gyros are utilized in this model as SMAP did not use any accelerometer data.)

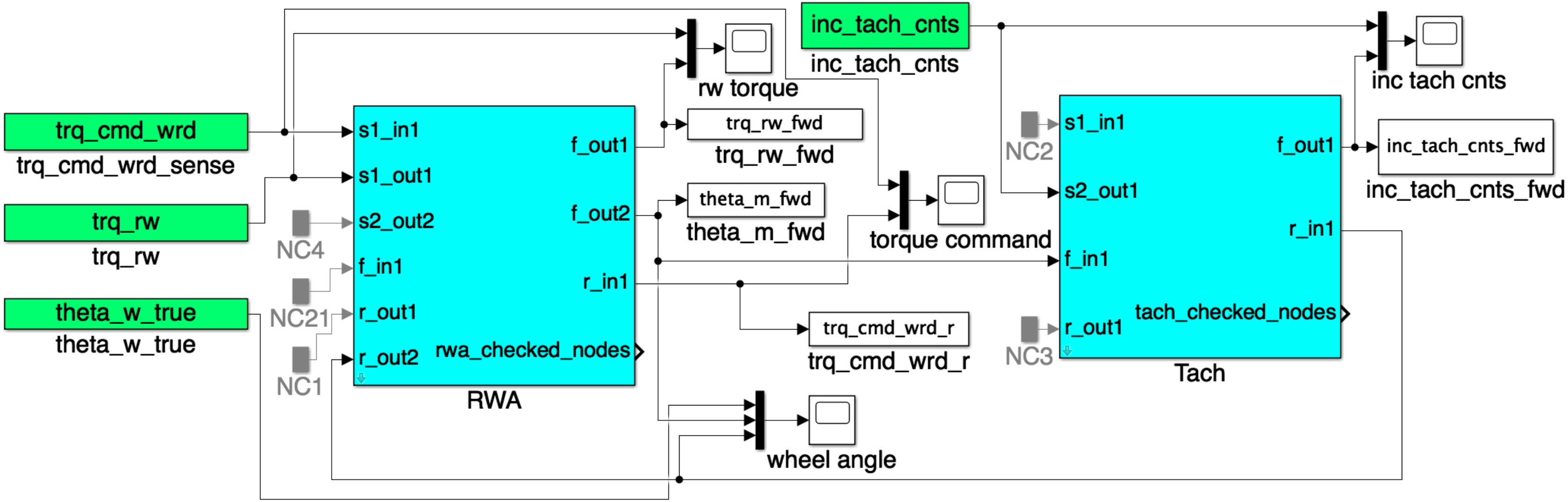


MONSID SRU Model

The MONSID SRU Model takes the true S/C attitude and rates as primary inputs. The Model outputs an attitude quaternion in J2000 to SRU frame and the angular rates at 8Hz.

MONSID GNC Models

Component Level Models : Actuators

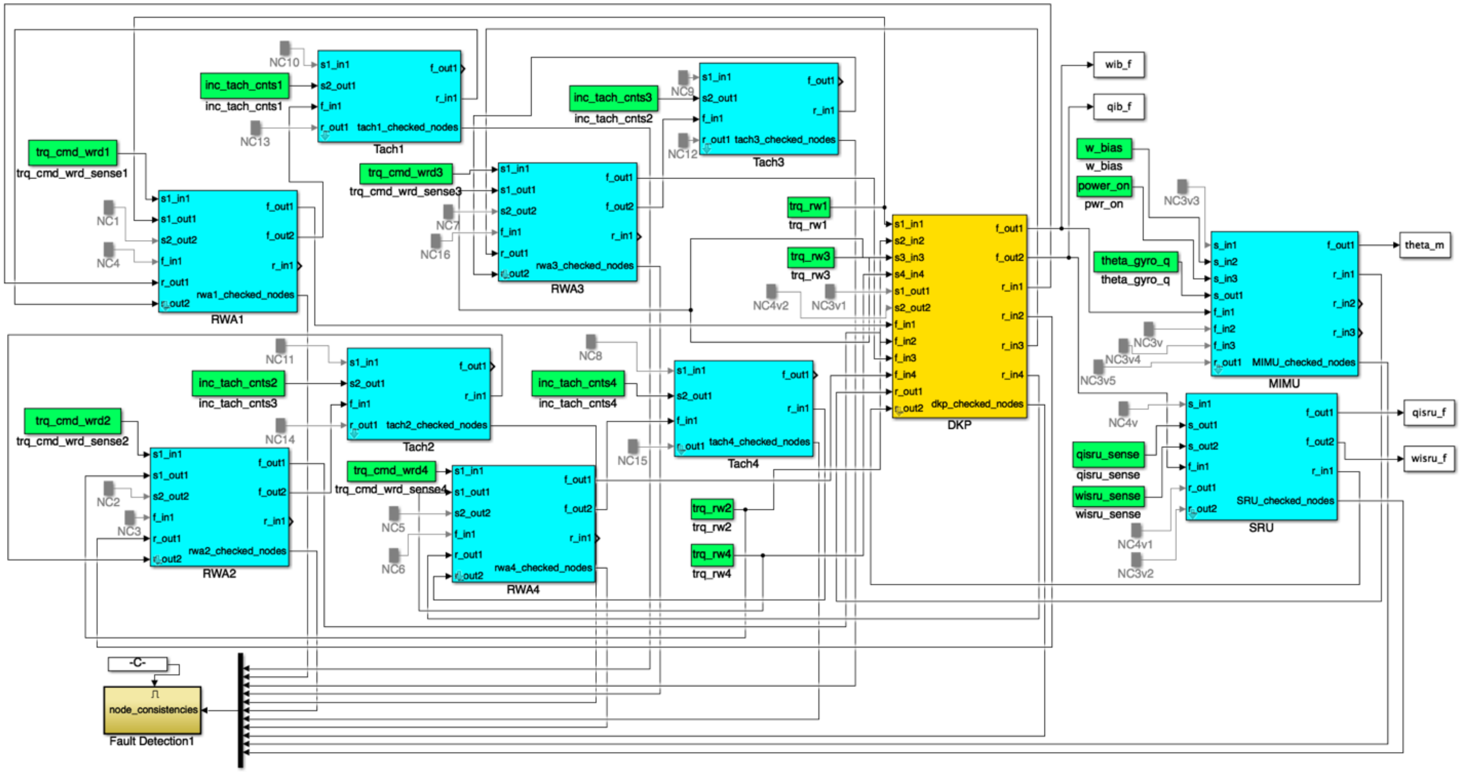


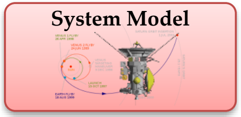
<u>MONSID RWA-TACH Model</u>

The MONSID RWA-TACH Model outputs the torque supplied to the spacecraft, the wheel angles and the corresponding tach count at 8Hz. A Torque command of specific duration is provided as one of the inputs to the model.

MONSID GNC Models

System Level Model

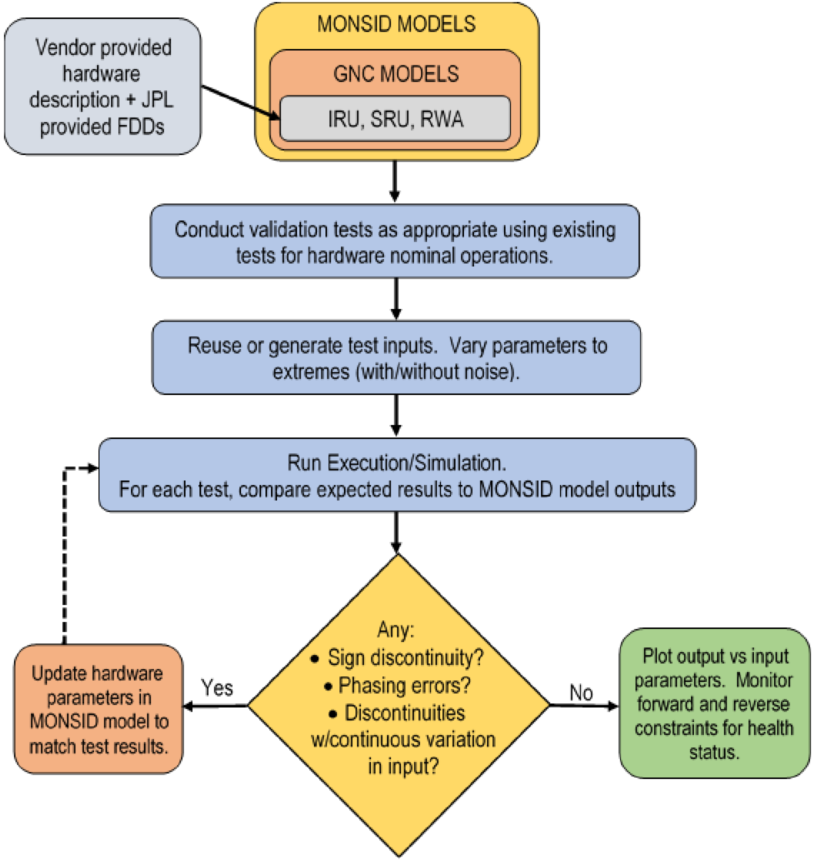




The **MONSID GNC System** model integrates the component models previously developed (MIMU, RWA, SRU). The **Dynamics Pseudocomponent (DKP)** component implements a simple physics model of the spacecraft dynamic environment during mission operations. It provides linkage between the outputs of the actuator and the inputs of the sensor.

Model Correctness

Established a Methodology for conducting V&V tests on the models



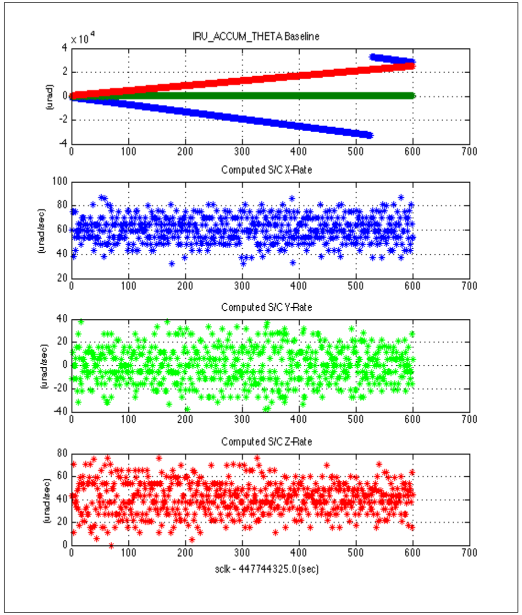
Vendor provided hardware description + JPL provided FDDs
MONSID MODELS
GNC MODELS
IRU, SRU, RWA
Conduct validation tests as appropriate using existing tests for hardware nominal operations.
Reuse or generate test inputs. Vary parameters to extremes (with/without noise).
Run Execution/Simulation.
For each test, compare expected results to MONSID model outputs
Any:
• Sign discontinuity?
• Phasing errors?
• Discontinuities w/continuous variation in input?
Yes
Update hardware parameters in MONSID model to match test results.
No
Plot output vs input parameters. Monitor forward and reverse constraints for health status.

Model Correctness

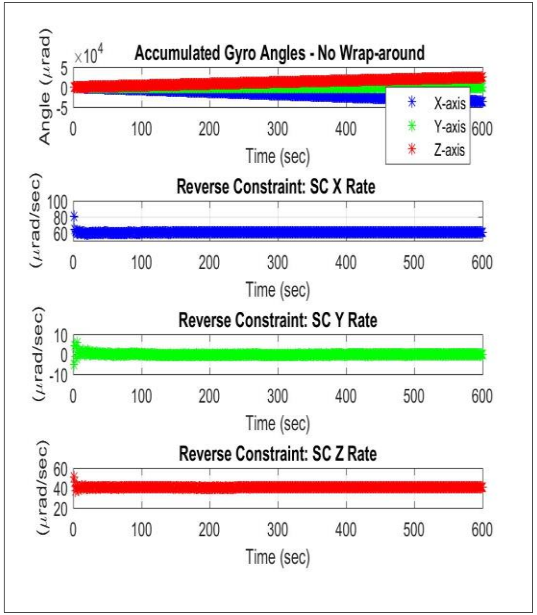
Component Level V&V Tests

MIMU Phasing Test

Determine whether the IRU could sense known Earth rotation rates in different orientations.



Expected Earth Rates from SMAP test procedure

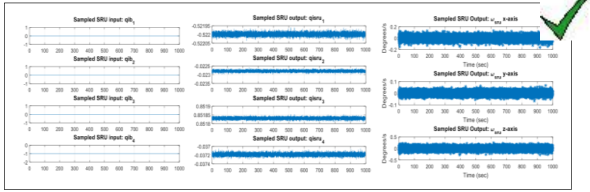


Earth Rates as simulated by MONSID MIMU Model

SRU Functional Test

Verify that the SRU supplies correct quaternion output when provided with some true initial attitude data.

Attitude Quaternion and Body Rates as simulated by MONSID SRU Model



RWA Functional Test

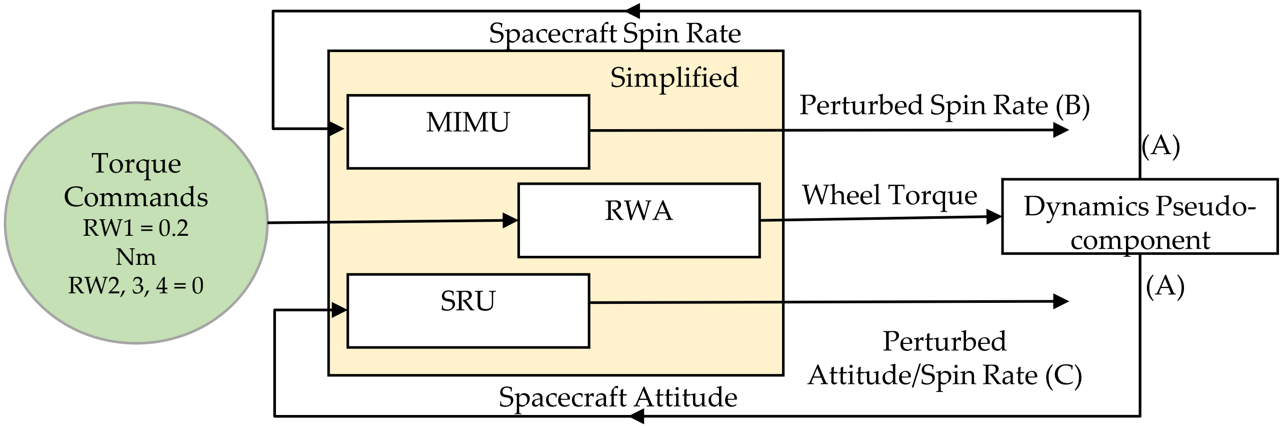
Send torque commands to the RWA for specified time duration and then verify the final tachometer count.

Tach Counts as simulated by MONSID RWA/TACH Model

Test Case	Torque Command	Total Duration	Expected Tach Count	MONSID Generated Tach Count
JPL provided flight hardware test: Test case 1	0.5 V(initial spin to wheel spin down)	25 sec	36 tach	**23 tach**
Supplier provided hardware acceptance test: Test case 1	V(high rate spin)	622.4 sec	1139.3 tach/sec	**1138.2 tach/sec**
Supplier provided hardware acceptance test: Test case 2	V(high rate spin)	955.2 sec	1738.5 tach/sec	**1737.0 tach/sec**

Model Correctness

System Level V&V Tests



Spacecraft Spin Rate

Simplified

Torque Commands
RW1 = 0.2 Nm
RW2, 3, 4 = 0

MIMU

RWA

SRU

Perturbed Spin Rate (B)

Wheel Torque

Dynamics Pseudo-component

(A)

(A)

Perturbed Attitude/Spin Rate (C)

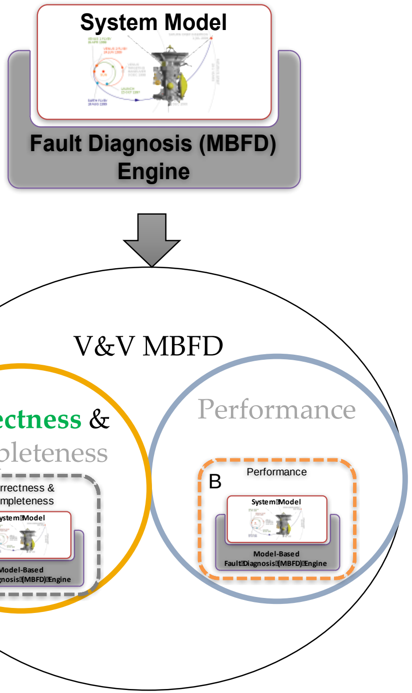
Spacecraft Attitude

End to End Validation Test performed on the GNC MONSID model to ensure Model Correctness

Time Point t	Expected Body rates	(A) "True" spacecraft body rates provided by DKP (forward values)	(B) MIMU measured body rates (forward values)	(C) SRU measured body rates (forward values)
10 sec	[0,-0.00028 , -0.00198] rad/sec	[-5.71E-06, -0.000277536, -0.001982401] rad/sec	[0, -0.00028, -0.00198] rad/sec ✓	[0, -0.0003, -0.0020] rad/sec ✓

Model Correctness

Results

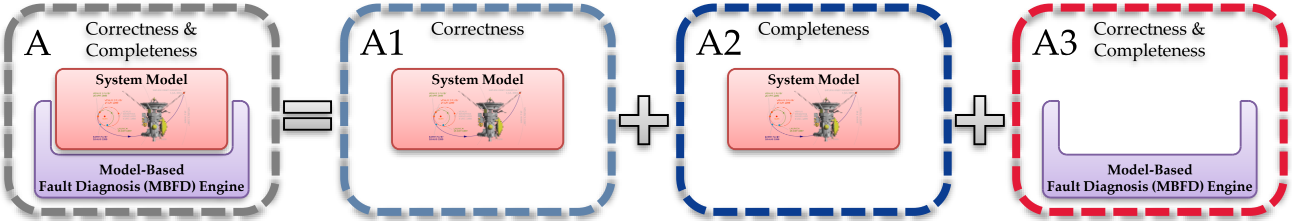

System Model
Fault Diagnosis (MBFD) Engine
V&V MBFD
Correctness & Completeness
Performance
A
Correctness & Completeness
System Model
Model-Based Fault Diagnosis (MBFD) Engine
B
Performance
System Model
Model-Based Fault Diagnosis (MBFD) Engine

- Successful demonstration of the feasibility to adapt the existing hardware test procedures to evaluate model correctness
- Ability to observe the nominal functionality and behavior of GNC hardware using component and system level model simulations
- Improved Confidence of the V&V Methodology to conduct further off-nominal behavioral test on the models

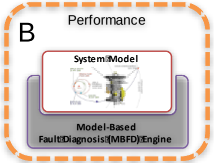
Future Work

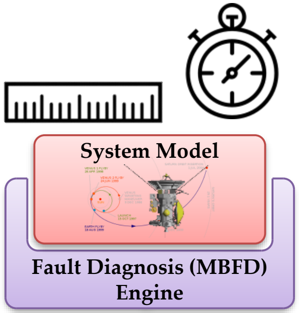
Further continuation of demonstrating assurance techniques…

MBFD Correctness and Completeness:

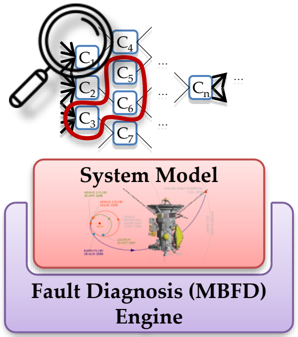

A
Correctness & Completeness
System Model
Model-Based Fault Diagnosis (MBFD) Engine
A1
Correctness
System Model
A2
Completeness
System Model
A3
Correctness & Completeness
Model-Based Fault Diagnosis (MBFD) Engine

MBFD Performance Analysis:

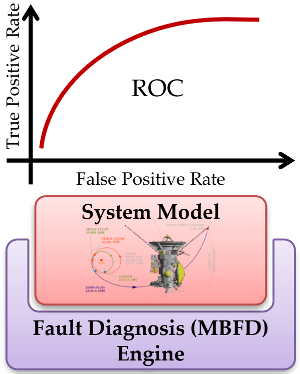




Gauge feasibility for onboard use by evaluating code size and speed



Determine how finely it can diagnose



Establish False Positive & False Negative Rates

Future work

Coming up in 2018…

- Perform static analysis on the MONSID diagnosis engine and report on test results and coverage analysis.
- Develop and demonstrate formal verification techniques for model correctness checking.
- Perform false-positive and false-negative diagnosis rate analysis.
- Refinements to MONSID to handle situations in which a faulty component could not be unambiguously identified.

Key Authors

Dr. Allen Nikora – NASA JPL

- *Software Reliability Engineer*

Dr. Seung Chung – NASA JPL

- *PI, manager*

Dr. Ksenia Kolcio – Okean Solutions

- *Consultant, MBFD supplier*

Dr. Lorraine Fesq – NASA JPL

- *MBFD Lead*

Priyanka Srivastava – NASA JPL

- *System V&V Engineer*

Thank you AeroConf 2018!

Jet Propulsion Laboratory
California Institute of Technology

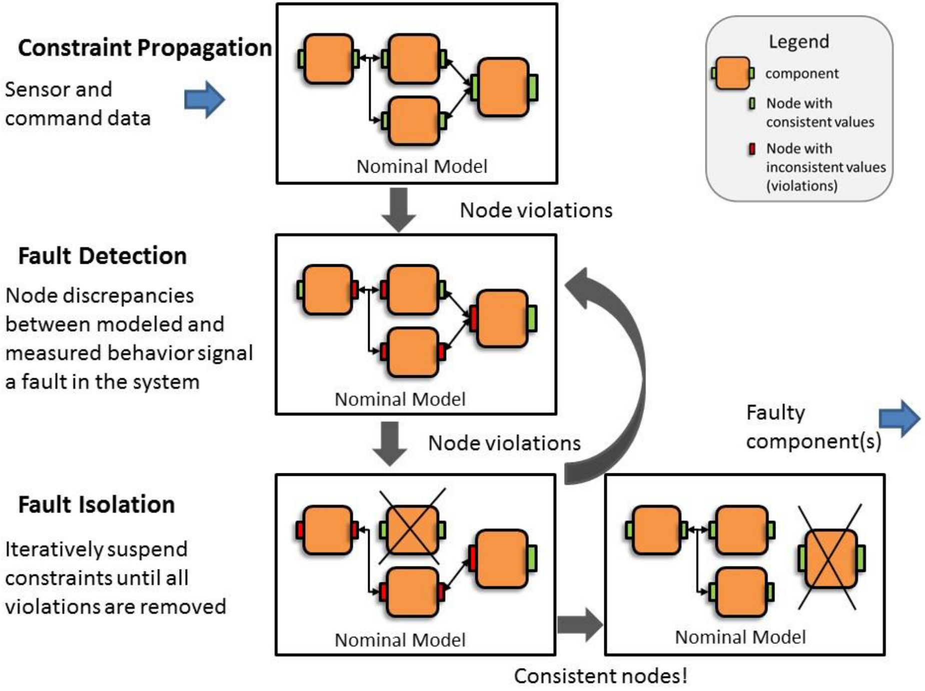
jpl.nasa.gov

Any Questions / Snide Remarks?

Backup Slides

Model-Based Fault Diagnosis Techniques

Constraint Suspension



Constraint Propagation
Sensor and command data
Nominal Model
Node violations
Fault Detection
Node discrepancies between modeled and measured behavior signal a fault in the system
Nominal Model
Node violations
Fault Isolation
Iteratively suspend constraints until all violations are removed
Nominal Model
Nominal Model
Consistent nodes!
Faulty component(s)
Legend
component
Node with consistent values
Node with inconsistent values (violations)

Model-based fault detection and isolation checks system data consistency at component nodes and systematically determines which component caused the off-nominal condition.